19

# PATENT ABSTRACTS OF JAPAN


(11)Publication number : 07-129373

(43)Date of publication of application : 19.05.1995

(51)Int.Cl. G06F 9/06
G06F 9/445
G06F 12/00

(21)Application number : 05-271826

(22)Date of filing : 29.10.1993



(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(72)Inventor : ONO HIDEKI
MATSUZAWA TOMOKO


(54) APPLICATION VERSION MANAGING DEVICE


(57)Abstract:

PURPOSE: To manage the consistency of versions between server application and a client application in an integrated/dispersed application system and to provide the application of the newest version to an application user.

CONSTITUTION: A server application version managing device 101 executes version inspection for a version inspection request outputted from a client application version managing device 104 at an application prepared date. At the time of detecting the discrepancy of versionsthe application of the newest version is automatically loaded down from a server system 100 to a client system 103so that the system 103 can store always the newest application.


CLAIMS

[Claim(s)]

[Claim 1]An application version control device comprising:

An application definition file with version information of application.

A server application version management means to manage the date and time of creation of application as a version.

A client application version management means to inspect a version of client application to an acknowledge request of a version from a serverand to notify a server of a result.

[Claim 2]The application version control device comprising according to claim 1:

A server application version management means to transmit to a client application upgraded when a version of application on a client system was old.
A client application version management means to update application of an old version which receives client application transmitted by server and exists on a client.

[Claim 3]The application version control device comprising according to claim 1: An application definition file holding version update information of application.
Whether renewal of a version of application which it is called from a manager program which has managed an integrated distributed applicationand is going to start was carried out normallyand a means to inspect.
A means to notify version disagreement to an application user and a manager program when renewal of a version is not carried out as a result of an inspection.

---

DETAILED DESCRIPTION

---

[Detailed Description of the Invention]
[0001]
[Industrial Application]This invention relates to the application version control device in integrated distribution application system.
[0002]
[Description of the Prior Art]In recent yearsconstruction of distributed application system has generalized application for the spread of LAN (Local Area Network)a substantial data base access tooletc. to twist to a server client system. When a system is built with a server client systemit is necessary to take the compatibility of a version between server application and client application.
[0003]The conventional application version control device is explained below. In drawing 13server application and 131 130 Client applicationThe message processing part in which 132 processes a network and 13a processes the wording of a telegram from client applicationThe wording-of-a-telegram preparing part in which 13b creates the wording of a telegram to client applicationThe version inspection section which inspects the version to which 13c has been sent from client application13 d The version information of client application13eThe transmission and reception section which transmits and receives 13 f of networks and informationthe message processing part which processes 13 g of wording of a telegram from server applicationThe command wording-of-a-telegram preparing part which creates the wording of a telegram which transmits to server application 13 hand 13i are the version information set up at the time of client application creation.
[0004]About the application version control device constituted as mentioned abovethe operation is explained below.
[0005]The client application 131 defines the version information 13i in a source

code at the time of application creationand creates a load module. When receiving service of server applicationcommand wording of a telegram is transmitted to the server application 130 through the network 132. It command wording-of-a-telegram preparing-part 13h Sets at this timethe self version information 13i defined in the load module is set as the version field of wording of a telegramand it transmits to the server application 130 through the wording-of-a-telegram transmission and reception section 13f. This is performed whenever client application transmits wording of a telegram to server application.
[0006]On the other handthe server application 130 also defines 13 d of client version information of the client application 131 in a source codeand creates a load module. The command wording of a telegram from the client application 131 which received by the transmission and reception section 13e is passed to the version inspection section 13c. The version inspection section 13c compares 13 d of client version information which the server application 130 holds with the version information set as command wording of a telegramWhen the disagreement of a version is detectederror wording-of-a-telegram transmission of "version disagreement" is requested from the wording-of-a-telegram preparing part 13band the service to the client application 131 is suspended. When the compatibility of the version can be takenwording of a telegram is passed to the message processing part 13aand the service to the client application 131 is continued.
[0007]When "version disagreement" wording of a telegram is detected from the server application device 130the client application 131 notifies the disagreement of a version to an application userand ends application service. All of these version managemnent processings are realized by the processing inside application.
[0008]
[Problem(s) to be Solved by the Invention]Howeverby the above-mentioned conventional methodsince confirming processing of version information was carried out by the wording of a telegram between the server application after application startingand client applicationif application was not startedit had the problem that the check of a version could not be performed.
[0009].Since version management is carried out inside application using the information currently held statically. In the integrated distribution application system which comprises two or more server applications and client applications. It also had the problem that there was a danger of creating application by the version information which had to give the information on version management to each applicationand was mistaken at the time of renewal of a version.
[0010]Version management is carried out without this invention's solving the above-mentioned conventional problemand being dependent on applicationand it aims at providing an application user with the application version control device which realizes the newest application offer.
[0011]
[Means for Solving the Problem]In order to attain this purpose. A server application version control device with which an application version control device of this invention manages a version of application on a serverand a client

application version control device which manages a version of application on a client. It comprises an application definition file which described an application name which is the target of version management.
[0012]
[Function]By this compositionan application controlling device is started from the manager program which has managed each application of integrated distribution application systemBy comparing the last date and time of creation currently held to the application definition file with the date and time of creation of the client application stored in the disk unit of a server system. It becomes manageable [ a version ] in the form where judged renewal of application and it became independent of application. Application can be used without a user being conscious of version management by starting automatically the application manager after application version management processing implementation. FurthermoreWhen a server application version control device detects the disagreement of a versionwith a download function and an update function. An application user can be provided with the newest application by being alike and downloading the newest application module to a client system.
[0013]
[Example]
(Example 1) It explains hereafterreferring to drawings for one example of this invention. Drawing 1 is an outline lineblock diagram of the integrated distribution application system provided with the application version control device in one example of this invention.
[0014]The server system in which 100 held server application and client application in drawing 1A server application version control device in which 101 manages the version of client application by the demand from a client systemA disk unit in which 102 holds server application and client application103the client system with which the client application operates 106104a client application version control device with which 107 manages the version of client application105the disk unit in which 108 holds client application and an application definition fileand 109 are networks which connect a server system and a client system.
[0015]Drawing 2 is an internal configuration figure of a server and the application version control device of a client. A disk unit in which 200 holds client application in drawing 2The version comparing element with which 201 compares the date and time of creationthe message processing part in which 202 processes the wording of a telegram from a client systemA transmission and reception section to which 203 performs communications control with a networka disk unit in which 204 holds client application and an application definition fileThe message processing part in which 205 processes wording of a telegramthe file management section in which 206 manages the client application and the application definition file in a disk unitThe transmission and reception section to which 207 performs communications control with a networkand 208 are file name storing regions which store the file name from which version disagreement was detected.

[0016]Drawing 3 is the format of transmitting and receiving-between server system and client system wording of a telegram. In drawing 3300 is the version test request wording of a telegram which transmits to a server from a clientand a test request reply telegraphic message [ as opposed to version test request wording of a telegram in 301 ].
[0017]Furthermoredrawing 4 is a detail view of the application definition file which has described the application name which conducts a version inspection.
[0018]In the integrated distribution application system constituted as mentioned aboveversion check processing is explained with the flow chart of drawing 5 and drawing 6.
[0019]Integrated distribution application system comprises two or more applications. There is application called the manager program which performs authenticating processing which checks the user of such applicationsand starting processing of application. A user usually starts this manager program. A manager program starts the client application version control device 104 in the client system 103 promptly after starting. The date and time of creation which set the file management section 206 to the application name (drawing 4 appl001.exe) last time from the application definition file (drawing 4) when the client application version control device 104 was started (1993/11/23 in a similar manner.) 16:13:38 is read and the message processing part 205 is passed (flow charts 5 and 1). The message processing part 205 creates the version test request wording of a telegram 300 using the application name and the date and time of creationand transmits to a server system via the transmission and reception section (flow charts 5 and 2) 207 (flow charts 5 and 3).
[0020]On the other handthe message processing part 202 will pass the application name and the date and time of creation which are set as wording of a telegram to the version comparing element 201if the version test request wording of a telegram 300 is received from the transmission and reception section 203 (flow charts 6 and 1) (flow charts 6 and 2). The version comparing element 201 to the disk unit 200. The date and time of creation of the specified application name which is held. It judges whether the client application of the disk unit 200 was updated in quest of difference with the date and time of creation which acquired and was passed from the message processing part (flow charts 6 and 3) 202and version (flow charts 6 and 4) coincidence or version disagreement is notified to the message processing part 202. The message processing part 202 sets a version inspection result to a flag as a result of the test request reply telegraphic message 301and transmits to a client system via the transmission and reception section (flow charts 6 and 5) 203 (flow charts 6 and 6).
[0021]The message processing part 205 sets a flag to the inspection result field of an application definition fileas a result of being set as the wording of a telegramif the test request reply telegraphic message 301 is received from the transmission and reception section 207 (flow charts 5 and 4) (flow charts 5 and 5). Version inspection processing is carried out to all the applications specified as the application definition file.

[0022]According to this exampleas mentioned above. The date and time of creation at the time of application creation is used as version information. By carrying out by preparing the both sides of a server system and a client system the application version control device which manages the version of client applicationand putting version management in block. Even if it does not have processing of version management in the inside of applicationa risk of creating application by the version information carrying out version management of application made the mistake in becoming possible can be prevented.
[0023](Example 2) It explains hereafterreferring to drawings for the 2nd example of this invention. Drawing 7 is a figure showing the internal structure of an application version control device. The explanation which has given the same numerals to the same component as the 1st example shown in drawing 2and overlapped about these is omitted. The download part by which 700 transmits application to a client system in drawing 7and 701 are update parts which update the client application currently held with the client system with the application transmitted from the server system. Drawing 8 is the format of the wording of a telegram transmitted and received between a server system and a client system at the time of automatic-updating processing. The download request wording of a telegram to which 800 is transmitted in drawing 8 at the time of a download requestand 801 are the download wording of a telegram included the contents of the application to update.
[0024]In the application version control device constituted as mentioned aboveautomatic-updating processing of client application is explained with the flow chart of drawing 9 and drawing 10.
[0025]The message processing part 205 will call the update part 701if the old application of a version exists on the client system after version inspection processing of Example 1. The update part 701 acquires an application name from the file name storing region 208creates the download request wording of a telegram 800and transmits to a server system via the transmission and reception section (flow charts 9 and 1) 207. (Flow charts 9 and 2) The message processing part 202 will pass the application name set as wording of a telegram to the download part 700if the download request wording of a telegram 800 is received from the transmission and reception section 203. The download part 700 is held to the disk unit 200 by this application name. A certain client application is openedit reads one record at a time (flow charts 10 and 1)the download wording of a telegram 801 is createdand it transmits to a client system via the transmission and reception section (flow charts 10 and 2) 203 (flow charts 10 and 3). The sequence number is added to the download wording of a telegram 801and the record omission at the time of download is prevented.
[0026]The update part 701 will write download wording of a telegram in the client application with which the disk unit 204 correspondsif the download wording of a telegram 801 is received from the transmission and reception section 207 (flow charts 9 and 3). All the records are received. If normal termination of the update process of client application is carried outthe date and time of creation after updating in

the date-and-time-of-creation field of an application definition file will be set upthe contents of the result flag will be changed into version coincidenceand download processing of the following (flow charts 9 and 4) application will be carried out. When an update process carries out abnormal terminationa flag carries out the next download processing as a result of the date and time of creationwithout updating.

[0027]By providing the function which downloads the newest load modulewhen Example 1 detects the disagreement of the version on a client system as mentioned above according to this exampleBeing able to distribute the newest load module to a client system automaticallyan application user becomes possible [ always using application by the newest version ]without being conscious.

[0028](Example 3) It explains hereafterreferring to drawings for the 3rd example of this invention. Drawing 11 is a figure showing the internal structure of a client application version control device. Whether in drawing 11as for 111renewal of a version was normally carried out at the time of application startingthe application starting check part to inspectand 112 are error display parts which display the message of version disagreement to an application user.

[0029]In the application version control device constituted as mentioned aboveprocessing of the version compatibility inspection at the time of application starting is explained with the flow chart of drawing 12.

[0030]Each application which constitutes integrated distribution application system is started from a manager program by an application user's directions. A manager program calls the application starting check part 111before starting the specified application. Under the present circumstancesthe application starting check part 111 is passed by making the load module name of the application which starts into a parameter.

[0031]Judge that the application starting check part 111 is not the application for version management if the same application name does not exist in the application name field of an application definition fileand to a manager program control. It returns (flow charts 12 and 1). When an application name existsthe contents of the result flag are inspected. If a result flag is version coincidenceversion coincidence will be passed to a manager program as a return valueand control will be returned (flow charts 12 and 2). In the case of version disagreementan error message is displayed by the error display part 112and it notifies an application user of version disagreement (flow charts 12 and 3). If an error display is canceled by the application userthe application starting check part 111 will return version disagreement to a manager programand will return control (flow charts 12 and 4).

[0032]A manager program carries out specified application starting processingwhen version coincidence is returned as a return value of the application starting check part 111and when version disagreement is returnedit stops application starting processing.

[0033]When renewal of a version is not able to carry out normally by forming a means to notify version disagreement to an application user and a manager program as mentioned above based on the file holding version update

informationand the version update information according to this exampleIt becomes possible to prevent beforehand the obstacle [ ********** ] at the time of using application by the mistaken version.

[0034]

[Effect of the Invention]above -- this invention -- integrated distribution -- an application -- a version control device is prepared for the both sides of a server and a client in the environment of a KESHO client application system. Thereforeeven if it does not process version management separately inside applicationthe outstanding application version control device which can carry out version management of the whole system and can update the load module of a client automatically is realizable.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1]The schematic diagram of the distributed application system provided with the application version control device in the 1st example of this invention

[Drawing 2]The schematic diagram of the internal structure of the application version control device in the 1st example of this invention

[Drawing 3]The message format figure at the time of the version inspection processing in the 1st example of this invention

[Drawing 4]The schematic diagram of the application definition file in the example of this invention

[Drawing 5]The flow chart of the client application version control device in the 1st example of this invention

[Drawing 6]The flow chart of the server application version control device in the 1st example of this invention

[Drawing 7]The schematic diagram of the internal structure of the application version control device in the 2nd example of this invention

[Drawing 8]The message format figure at the time of the download processing in the 2nd example of this invention

[Drawing 9]The flow chart of the client application version control device in the 2nd example of this invention

[Drawing 10]The flow chart of the server application version control device in the 2nd example of this invention

[Drawing 11]The schematic diagram of the internal structure of the application version control device in the 3rd example of this invention

[Drawing 12]The flow chart of the application starting check processing in the 3rd example of this invention

[Drawing 13]The schematic diagram of the conventional application version control device

[Description of Notations]

100 Server system
101 Server application version control device
102 Disk unit
103 Client system
104 Client application version control device
105 Disk unit
106 Client system
107 Client application version control device
108 Disk unit
109 Network
200 Disk unit
201 Version comparing element
202 Message processing part
203 Transmission and reception section
204 Disk unit
205 Message processing part
206 File management section
207 Transmission and reception section
300 Version test request wording of a telegram
301 Test request reply telegraphic message
700 Download part
701 Update part
800 Download request wording of a telegram
801 Download wording of a telegram
111 Application starting check part
112 Error display part
130 Server application device
131 Client application device
132 Network
13a Message processing part
13b Wording-of-a-telegram preparing part
13c Version inspection section
13 d Client version information
13e Wording-of-a-telegram transmission and reception section
13 f Wording-of-a-telegram transmission and reception section
13 g Message processing part
13 h Command wording-of-a-telegram preparing part
13i Version information

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

**特開平7－129373**

(43)公開日　平成７年(1995)５月19日

| (51)Int.Cl.[6] | | 識別記号 | 庁内整理番号 | ＦＩ | 技術表示箇所 |
|---|---|---|---|---|---|
| Ｇ０６Ｆ | 9/06 | ４１０　Ｑ | 9367－5B | | |
| | 9/445 | | | | |
| | 12/00 | ５１７ | 8944－5B | | |
| | | | 9367－5B | Ｇ０６Ｆ　9/ 06　４２０　Ｍ | |

審査請求　未請求　請求項の数３　ＯＬ　（全 10 頁）

(21)出願番号　特願平5－271826

(22)出願日　平成５年(1993)10月29日

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者　小野　英樹
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　松澤　智子
大阪府門真市大字門真1006番地　松下電器産業株式会社内

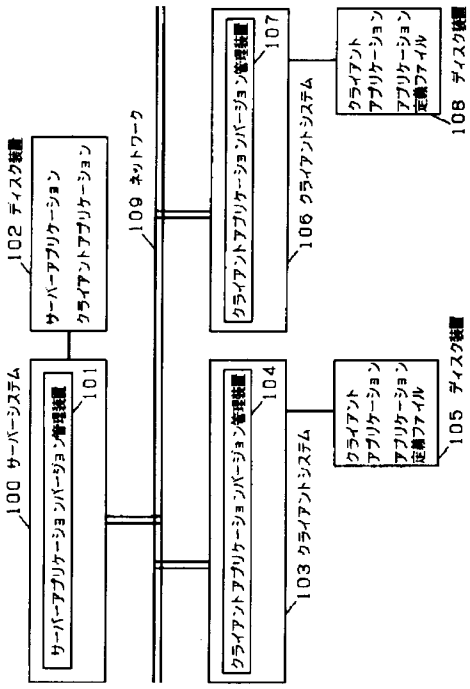
(74)代理人　弁理士　小鍛治　明　（外２名）

(54)【発明の名称】　アプリケーションバージョン管理装置

(57)【要約】

【目的】　統合分散アプリケーションシステムにおいてサーバーアプリケーションとクライアントアプリケーション間のバージョンの整合性を管理し、アプリケーション使用者に最新バージョンのアプリケーションを提供することを目的とする。

【構成】　サーバーアプリケーションバージョン管理装置１０１はクライアントアプリケーションバージョン管理装置１０４からのバージョン検査依頼に対してアプリケーション作成日付でバージョンの検査を実施する。バージョン不一致を検出するとサーバーシステム１００からクライアントシステム１０３へ最新バージョンのアプリケーションを自動的にダウンロードすることによりクライアントシステムは常に最新のアプリケーションを保持することが可能となる。



【特許請求の範囲】
【請求項１】　アプリケーションのバージョン情報を持つアプリケーション定義ファイルと、アプリケーションの作成日時をバージョンとして管理するサーバーアプリケーションバージョン管理手段と、サーバーからのバージョンの確認要求に対してクライアントアプリケーションのバージョンを検査してサーバーへ結果を通知するクライアントアプリケーションバージョン管理手段とを備えたアプリケーションバージョン管理装置。
【請求項２】　クライアントシステム上のアプリケーションのバージョンが古い場合バージョンアップされたアプリケーションをクライアントへ送信するサーバーアプリケーションバージョン管理手段と、サーバーから送信されてくるクライアントアプリケーションを受信してクライアント上に存在する古いバージョンのアプリケーションを更新するクライアントアプリケーションバージョン管理手段を備えた請求項１記載のアプリケーションバージョン管理装置。
【請求項３】　アプリケーションのバージョン更新情報を保持するアプリケーション定義ファイルと、統合分散アプリケーションを管理しているマネージャープログラムから呼び出され起動しようとしているアプリケーションのバージョン更新が正常に実施されたか検査する手段と、検査の結果バージョン更新が実施されていない場合アプリケーション使用者およびマネージャープログラムにバージョン不一致を通知する手段とを備えた請求項１記載のアプリケーションバージョン管理装置 。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は統合分散アプリケーションシステムにおけるアプリケーションバージョン管理装置に関するものである。
【０００２】
【従来の技術】近年、アプリケーションはＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）の普及、データベースアクセスツール等の充実によりサーバークライアント方式によるに分散アプリケーションシステムの構築が一般化している。サーバークライアント方式でシステムを構築した場合サーバーアプリケーションとクライアントアプリケーション間でバージョンの整合性をとる必要がある。
【０００３】以下に従来のアプリケーションバージョン管理装置について説明する。図１３において１３０はサーバーアプリケーション、１３１はクライアントアプリケーション、１３２はネットワーク、１３ａはクライアントアプリケーションからの電文を処理する電文処理部、１３ｂはクライアントアプリケーションへの電文を作成する電文作成部、１３ｃはクライアントアプリケーションから送られてきたバージョンを検査するバージョン検査部、１３ｄはクライアントアプリケーションのバージョン情報、１３ｅ、１３ｆはネットワークと情報を送受信する送受信部、１３ｇはサーバーアプリケーションからの電文を処理する電文処理部、１３ｈはサーバーアプリケーションへ送信する電文を作成するコマンド電文作成部、１３ｉはクライアントアプリケーション作成時に設定したバージョン情報である。
【０００４】以上のように構成されたアプリケーションバージョン管理装置について、以下にその動作について説明する。
【０００５】クライアントアプリケーション１３１はアプリケーション作成時にソースコード内にバージョン情報１３ｉを定義してロードモジュールを作成する。サーバーアプリケーションのサービスを受ける時ネットワーク１３２を通じてサーバーアプリケーション１３０へコマンド電文を送信する。この時コマンド電文作成部１３ｈおいて、電文のバージョンフィールドにロードモジュール内に定義してある自己のバージョン情報１３ｉを設定して電文送受信部１３ｆを通じてサーバーアプリケーション１３０へ送信する。これはクライアントアプリケーションがサーバーアプリケーションへ電文を送信する毎に行われる。
【０００６】一方サーバーアプリケーション１３０もクライアントアプリケーション１３１のクライアントバージョン情報１３ｄをソースコード内に定義してロードモジュールを作成する。送受信部１３ｅで受信したクライアントアプリケーション１３１からのコマンド電文をバージョン検査部１３ｃに渡す。バージョン検査部１３ｃはサーバーアプリケーション１３０が保持しているクライアントバージョン情報１３ｄとコマンド電文に設定されているバージョン情報を比較し、バージョンの不一致を検出した場合は電文作成部１３ｂに「バージョン不一致」のエラー電文送信を依頼してクライアントアプリケーション１３１に対するサービスを停止する。バージョンの整合性がとれている場合は電文処理部１３ａに電文を渡しクライアントアプリケーション１３１に対するサービスを継続する。
【０００７】クライアントアプリケーション１３１はサーバーアプリケーション装置１３０から「バージョン不一致」電文を検出した場合はアプリケーション使用者に対してバージョンの不一致を通知してアプリケーションサービスを終了する。これらのバージョン管理処理はすべてアプリケーション内部の処理によって実現されている。
【０００８】
【発明が解決しようとする課題】しかしながら上記の従来の方法ではアプリケーション起動後サーバーアプリケーションとクライアントアプリケーション間の電文によりバージョン情報の確認処理を実施しているためアプリケーションを起動しなければバージョンの確認ができないという問題点を有していた。

【０００９】また、アプリケーション内部に静的に保持している情報を使用してバージョン管理を実施しているため複数のサーバーアプリケーションとクライアントアプリケーションで構成されている統合分散アプリケーションシステムにおいて個々のアプリケーションにバージョン管理の情報を持たせなければならずバージョン更新時に誤ったバージョン情報でアプリケーションを作成する危険性があるという問題点も有していた。

【００１０】本発明は上記従来の問題点を解決するものでアプリケーションに依存せずにバージョン管理を実施し、アプリケーション使用者に最新のアプリケーション提供を実現するアプリケーションバージョン管理装置を提供することを目的とする。

【００１１】

【課題を解決するための手段】この目的を達成するために本発明のアプリケーションバージョン管理装置はサーバー上でアプリケーションのバージョンを管理するサーバーアプリケーションバージョン管理装置とクライアント上でアプリケーションのバージョンを管理するクライアントアプリケーションバージョン管理装置とバージョン管理の対象となるアプリケーション名を記述したアプリケーション定義ファイルから構成される。

【００１２】

【作用】この構成によって、アプリケーション管理装置は統合分散アプリケーションシステムの各アプリケーションを管理しているマネージャープログラムから起動され、アプリケーション定義ファイルに保持している前回の作成日時とサーバーシステムのディスク装置に格納しているクライアントアプリケーションの作成日時を比較することによりアプリケーションの更新を判断しアプリケーションから独立した形でバージョンの管理が可能となる。またアプリケーションバージョン管理処理実施後アプリケーションマネージャーを自動的に起動することにより使用者はバージョン管理を意識することなくアプリケーションを使用することができる。さらに、サーバーアプリケーションバージョン管理装置によりバージョンの不一致を検出した場合はダウンロード機能とアップデート機能によりにクライアントシステムへ最新アプリケーションモジュールをダウンロードすることによりアプリケーション使用者に最新のアプリケーションを提供することができる。

【００１３】

【実施例】

（実施例１）以下、本発明の一実施例について図面を参照しながら説明する。図１は本発明の一実施例におけるアプリケーションバージョン管理装置を備えた統合分散アプリケーションシステムの概略構成図である。

【００１４】図１において１００はサーバーアプリケーションとクライアントアプリケーションを保持したサーバーシステム、１０１はクライアントシステムからの要求によりクライアントアプリケーションのバージョンを管理するサーバーアプリケーションバージョン管理装置、１０２はサーバーアプリケーションとクライアントアプリケーションを保持するディスク装置、１０３、１０６はクライアントアプリケーションが動作するクライアントシステム、１０４、１０７はクライアントアプリケーションのバージョンを管理するクライアントアプリケーションバージョン管理装置、１０５、１０８はクライアントアプリケーションとアプリケーション定義ファイルを保持するディスク装置、１０９はサーバーシステムとクライアントシステムを接続するネットワークである。

【００１５】図２はサーバーおよびクライアントのアプリケーションバージョン管理装置の内部構成図である。図２においては２００はクライアントアプリケーションを保持しているディスク装置、２０１は作成日時を比較するバージョン比較部、２０２はクライアントシステムからの電文を処理する電文処理部、２０３はネットワークとの通信制御を行う送受信部、２０４はクライアントアプリケーションとアプリケーション定義ファイルを保持しているディスク装置、２０５は電文を処理する電文処理部、２０６はディスク装置内のクライアントアプリケーションとアプリケーション定義ファイルを管理するファイル管理部、２０７はネットワークとの通信制御を行う送受信部、２０８はバージョン不一致が検出されたファイル名を格納するファイル名格納領域である。

【００１６】図３はサーバーシステムとクライアントシステム間で送受信されるの電文のフォーマットである。図３において３００はクライアントからサーバーへ送信するバージョン検査依頼電文、３０１はバージョン検査依頼電文に対する検査依頼応答電文である。

【００１７】さらに図４はバージョン検査をするアプリケーション名を記述してあるアプリケーション定義ファイルの詳細図である。

【００１８】以上のように構成された統合分散アプリケーションシステムにおいてバージョンチェック処理について図５、図６のフローチャートと共に説明する。

【００１９】統合分散アプリケーションシステムは複数のアプリケーションから構成されている。これらのアプリケーションの使用者を確認する認証処理やアプリケーションの起動処理を行うマネージャープログラムと呼ばれるアプリケーションがある。使用者は通常このマネージャープログラムを起動する。マネージャープログラムは起動後直ちにクライアントシステム１０３内のクライアントアプリケーションバージョン管理装置１０４を起動する。クライアントアプリケーションバージョン管理装置１０４が起動されるとファイル管理部２０６はアプリケーション定義ファイル（図４）からアプリケーション名（図４ではapp1001.exe）と前回設定した作成日時（同様に1993/11/23 16:13:38）を読み電文処理部２０

５へ渡す（ﾌﾛｰﾁｬｰﾄ５・１）。電文処理部２０５は同アプリケーション名と作成日時を使用してバージョン検査依頼電文３００を作成し（ﾌﾛｰﾁｬｰﾄ５・２）送受信部２０７を介してサーバーシステムへ送信する（ﾌﾛｰﾁｬｰﾄ５・３）。

【００２０】一方電文処理部２０２はバージョン検査依頼電文３００を送受信部２０３から受信すると（ﾌﾛｰﾁｬｰﾄ６・１）電文に設定されているアプリケーション名と作成日時をバージョン比較部２０１へ渡す（ﾌﾛｰﾁｬｰﾄ６・２）。バージョン比較部２０１はディスク装置２００に保持している指定されたアプリケーション名の作成日時を取得し（ﾌﾛｰﾁｬｰﾄ６・３）電文処理部２０２から渡された作成日時との差分を求めディスク装置２００のクライアントアプリケーションが更新されたか判断し（ﾌﾛｰﾁｬｰﾄ６・４）バージョン一致またはバージョン不一致を電文処理部２０２へ通知する。電文処理部２０２はバージョン検査結果を検査依頼応答電文３０１の結果フラグにセットして（ﾌﾛｰﾁｬｰﾄ６・５）送受信部２０３を介してクライアントシステムへ送信する（ﾌﾛｰﾁｬｰﾄ６・６）。

【００２１】電文処理部２０５は送受信部２０７から検査依頼応答電文３０１を受信すると（ﾌﾛｰﾁｬｰﾄ５・４）同電文に設定されている結果フラグをアプリケーション定義ファイルの検査結果フィールドにセットする（ﾌﾛｰﾁｬｰﾄ５・５）。バージョン検査処理はアプリケーション定義ファイルに指定されているすべてのアプリケーションに対して実施する。

【００２２】以上のように本実施例によれば、アプリケーション作成時の作成日時をバージョン情報として利用してクライアントアプリケーションのバージョンを管理するアプリケーションバージョン管理装置をサーバーシステムとクライアントシステムの双方に設けてバージョン管理を一括して行うことによりアプリケーション内部にバージョン管理の処理を持たなくてもアプリケーションのバージョン管理を実施することが可能となり誤ったバージョン情報でアプリケーションを作成してしまうという危険を防止することができる。

【００２３】（実施例２）以下、本発明の第２の実施例について図面を参照しながら説明する。図７はアプリケーションバージョン管理装置の内部構造を示す図である。図２に示した第１の実施例と同じ構成要素には同じ符号が付与してありこれらについては重複した説明を省略する。図７において７００はクライアントシステムへアプリケーションを送信するダウンロード部、７０１はサーバーシステムから送信されてきたアプリケーションでクライアントシステムで保持しているクライアントアプリケーションを更新するアップデート部である。また図８は自動更新処理時にサーバーシステムとクライアントシステム間で送受信される電文のフォーマットである。図８において８００はダウンロード要求時に送信されるダウンロード依頼電文、８０１は更新するアプリケーションの内容を含んだダウンロード電文である。

【００２４】以上のように構成されたアプリケーションバージョン管理装置においてクライアントアプリケーションの自動更新処理について図９、図１０のフローチャートと共に説明する。

【００２５】電文処理部２０５は実施例１のバージョン検査処理後クライアントシステム上にバージョンの古いアプリケーションが存在するとアップデート部７０１を呼出す。アップデート部７０１はファイル名格納領域２０８からアプリケーション名を取得してダウンロード依頼電文８００を作成し（ﾌﾛｰﾁｬｰﾄ９・１）送受信部２０７を介してサーバーシステムへ送信する。（ﾌﾛｰﾁｬｰﾄ９・２）電文処理部２０２は送受信部２０３からダウンロード依頼電文８００を受信すると電文に設定されているアプリケーション名をダウンロード部７００へ渡す。ダウンロード部７００はこのアプリケーション名でディスク装置２００に保持してあるクライアントアプリケーションをオープンして１レコードずつ読み込み（ﾌﾛｰﾁｬｰﾄ１０・１）ダウンロード電文８０１を作成し（ﾌﾛｰﾁｬｰﾄ１０・２）送受信部２０３を介してクライアントシステムへ送信する（ﾌﾛｰﾁｬｰﾄ１０・３）。ダウンロード電文８０１には順序番号を付加しダウンロード時のレコード抜けを防止する。

【００２６】アップデート部７０１は送受信部２０７からダウンロード電文８０１を受信するとディスク装置２０４の該当するクライアントアプリケーションにダウンロード電文を書込む（ﾌﾛｰﾁｬｰﾄ９・３）。すべてのレコードを受信してクライアントアプリケーションの更新処理を正常終了するとアプリケーション定義ファイルの作成日時フィールドに更新後の作成日時を設定し結果フラグの内容をバージョン一致に変更し（ﾌﾛｰﾁｬｰﾄ９・４）次のアプリケーションのダウンロード処理を実施する。更新処理が異常終了した場合は作成日時、結果フラグは更新せずに次のダウンロード処理を実施する。

【００２７】以上のように本実施例によれば、実施例１によりクライアントシステム上のバージョンの不一致を検出した場合、最新ロードモジュールをダウンロードする機能を設けることにより、自動的にクライアントシステムに最新ロードモジュールを配布することができ、アプリケーション使用者は意識する事なく常に最新のバージョンでアプリケーションを利用することが可能となる。

【００２８】（実施例３）以下、本発明の第３の実施例について図面を参照しながら説明する。図１１はクライアントアプリケーションバージョン管理装置の内部構造を示す図である。図１１において１１１はアプリケーション起動時にバージョン更新が正常に実施されたか検査するアプリケーション起動チェック部、１１２はアプリケーション使用者に対してバージョン不一致のメッセー

ジを表示するエラー表示部である。

【００２９】以上のように構成されたアプリケーションバージョン管理装置においてアプリケーション起動時のバージョン整合性検査の処理について図１２のフローチャートと共に説明する。

【００３０】統合分散アプリケーションシステムを構成している各アプリケーションはアプリケーション使用者の指示によりマネージャープログラムから起動される。マネージャープログラムは指定されたアプリケーションを起動する前にアプリケーション起動チェック部１１１を呼び出す。この際起動するアプリケーションのロードモジュール名をパラメータとしてアプリケーション起動チェック部１１１に渡す。

【００３１】アプリケーション起動チェック部１１１はアプリケーション定義ファイルのアプリケーション名フィールドに同一のアプリケーション名が存在しなければバージョン管理対象のアプリケーションではないと判断しマネージャープログラムに制御を戻す（ﾌﾛｰﾁｬｰﾄ１２・１）。アプリケーション名が存在する場合は結果フラグの内容を検査する。結果フラグがバージョン一致ならばマネージャープログラムにバージョン一致を返値として渡し制御を戻す（ﾌﾛｰﾁｬｰﾄ１２・２）。バージョン不一致の場合はエラー表示部１１２によりエラーメッセージを表示してアプリケーション使用者にバージョン不一致を通知する（ﾌﾛｰﾁｬｰﾄ１２・３）。アプリケーション使用者によりエラー表示が解除されるとアプリケーション起動チェック部１１１はマネージャープログラムにバージョン不一致を返して制御を戻す（ﾌﾛｰﾁｬｰﾄ１２・４）。

【００３２】マネージャープログラムはアプリケーション起動チェック部１１１の返値としてバージョン一致が返された場合は指定されたアプリケーション起動処理を実施し、バージョン不一致が返された場合はアプリケーション起動処理を中止する。

【００３３】以上のように本実施例によれば、バージョン更新情報を保持するファイルと同バージョン更新情報を基にアプリケーション使用者およびマネージャープログラムにバージョン不一致を通知する手段を設けることによりバージョン更新が正常に実施できなかった場合、誤ったバージョンでアプリケーションを使用した際の予測不可能な障害を未然に防止することが可能となる。

【００３４】

【発明の効果】以上のように本発明は統合分散アプリケーショクライアントアプリケーションシステムの環境においてサーバー、クライアントの双方にバージョン管理装置を設けることによりアプリケーション内部で個々にバージョン管理の処理を行わなくてもシステム全体のバージョン管理を実施しかつクライアントのロードモジュールを自動的に更新することができる優れたアプリケーションバージョン管理装置を実現できるものである。

【図面の簡単な説明】

【図１】本発明の第１の実施例におけるアプリケーションバージョン管理装置を備えた分散アプリケーションシステムの概要図

【図２】本発明の第１の実施例におけるアプリケーションバージョン管理装置の内部構造の概要図

【図３】本発明の第１の実施例におけるバージョン検査処理時の電文フォーマット図

【図４】本発明の実施例におけるアプリケーション定義ファイルの概要図

【図５】本発明の第１の実施例におけるクライアントアプリケーションバージョン管理装置のフローチャート

【図６】本発明の第１の実施例におけるサーバーアプリケーションバージョン管理装置のフローチャート

【図７】本発明の第２の実施例におけるアプリケーションバージョン管理装置の内部構造の概要図

【図８】本発明の第２の実施例におけるダウンロード処理時の電文フォーマット図

【図９】本発明の第２の実施例におけるクライアントアプリケーションバージョン管理装置のフローチャート

【図１０】本発明の第２の実施例におけるサーバーアプリケーションバージョン管理装置のフローチャート

【図１１】本発明の第３の実施例におけるアプリケーションバージョン管理装置の内部構造の概要図

【図１２】本発明の第３の実施例におけるアプリケーション起動チェック処理のフローチャート

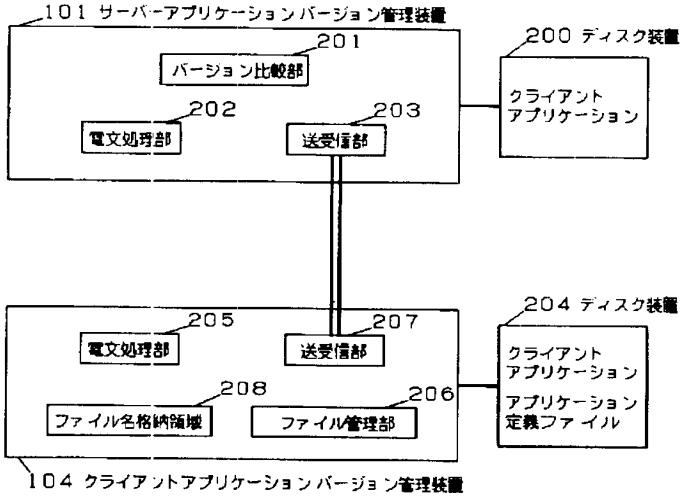
【図１３】従来のアプリケーションバージョン管理装置の概要図

【符号の説明】

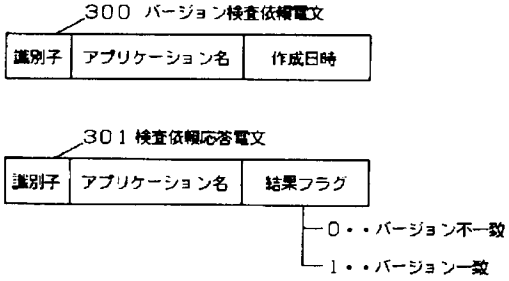
１００　サーバーシステム
１０１　サーバーアプリケーションバージョン管理装置
１０２　ディスク装置
１０３　クライアントシステム
１０４　クライアントアプリケーションバージョン管理装置
１０５　ディスク装置
１０６　クライアントシステム
１０７　クライアントアプリケーションバージョン管理装置
１０８　ディスク装置
１０９　ネットワーク
２００　ディスク装置
２０１　バージョン比較部
２０２　電文処理部
２０３　送受信部
２０４　ディスク装置
２０５　電文処理部
２０６　ファイル管理部
２０７　送受信部
３００　バージョン検査依頼電文

３０１　検査依頼応答電文
７００　ダウンコード部
７０１　アップデート部
８００　ダウンコード依頼電文
８０１　ダウンコード電文
１１１　アプリケーション起動チェック部
１１２　エラー表示部
１３０　サーバーアプリケーション装置
１３１　クライアントアプリケーション装置
１３２　ネットワーク
１３ａ　電文処理部
１３ｂ　電文作成部
１３ｃ　バージョン検査部
１３ｄ　クライアントバージョン情報
１３ｅ　電文送受信部
１３ｆ　電文送受信部
１３ｇ　電文処理部
１３ｈ　コマンド電文作成部
１３ｉ　バージョン情報

【図２】




【図３】

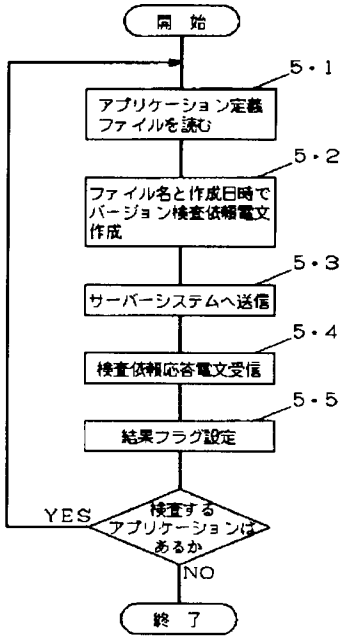



【図４】

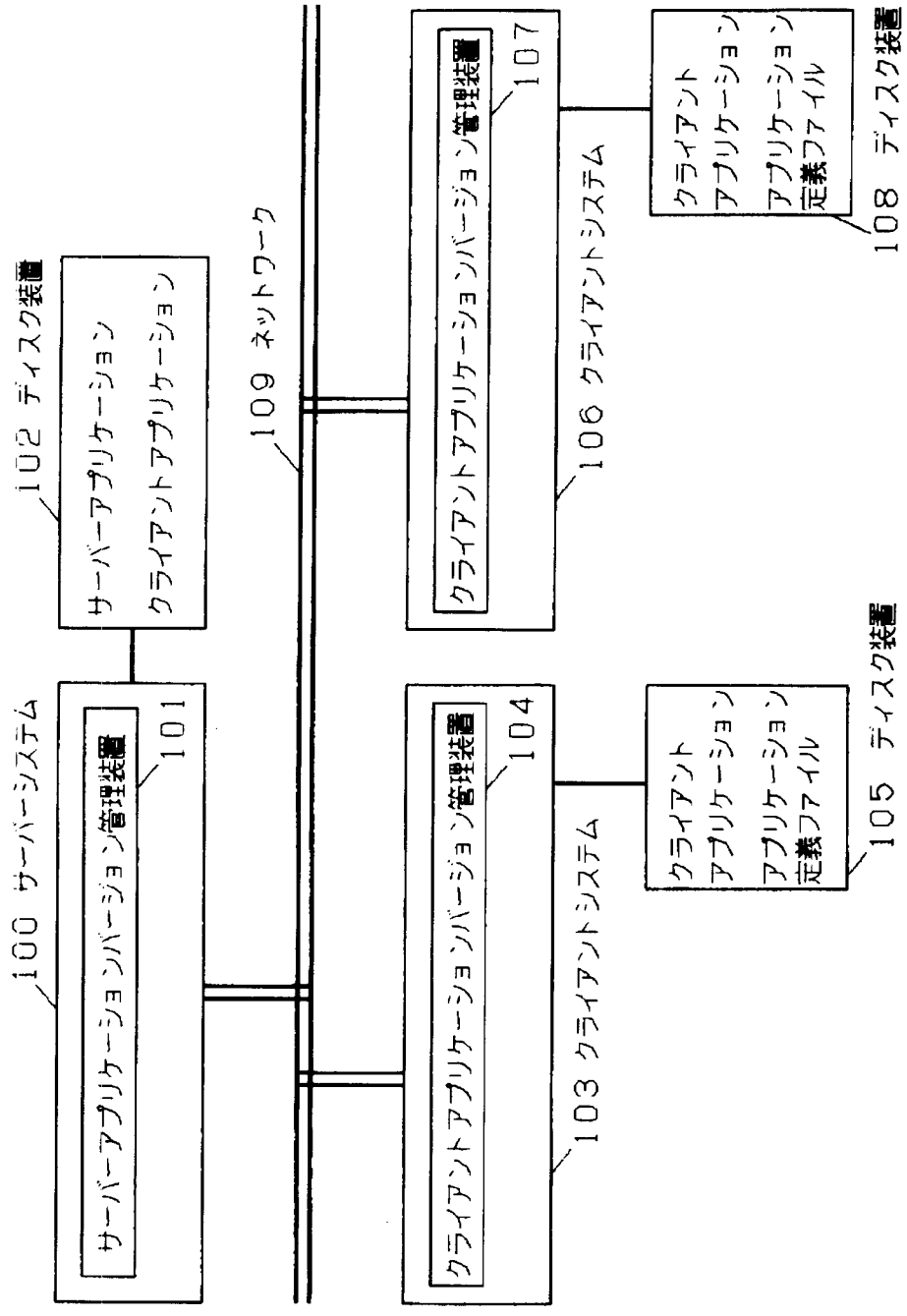
| アプリケーション名 | 作成日時 | 結果フラグ |
|---|---|---|
| appl001.exe | 1993/11/23 16:13:38 | 1 |
| appl002.exe | 1993/11/23 17:56:00 | 0 |
| ・ | ・ | ・ |
| ・ | ・ | ・ |
| appl009.exe | 1993/11/23 16:13:38 | 0 |

0・・バージョン不一致
1・・バージョン一致

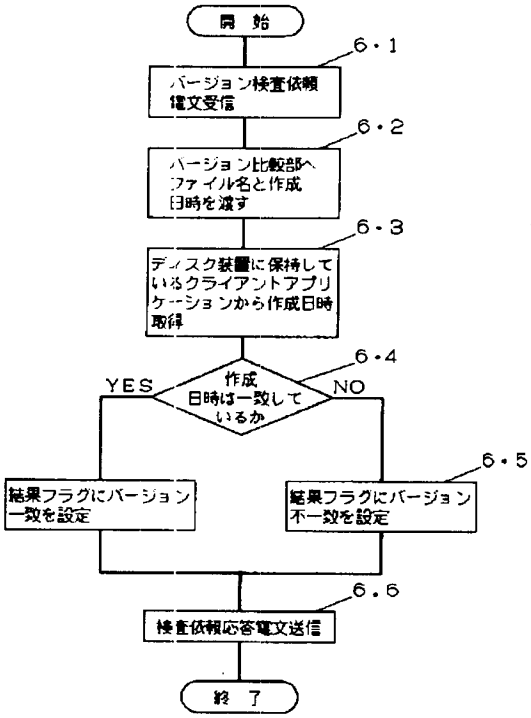
【図５】



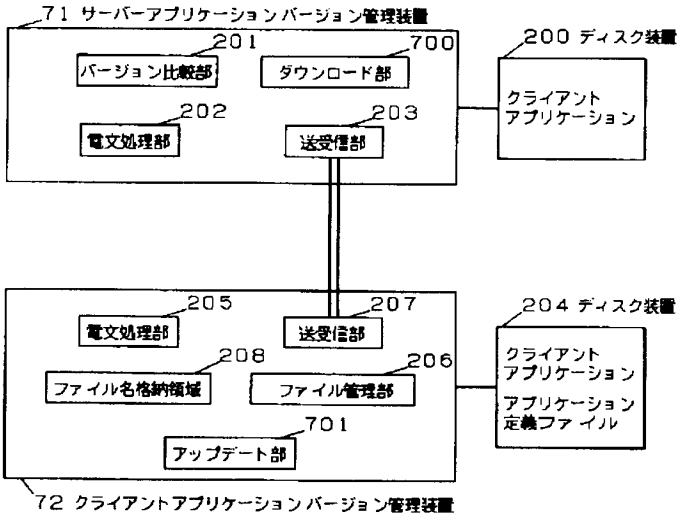
【図1】


100 サーバーシステム
101
サーバーアプリケーションバージョン管理装置
102 ディスク装置
サーバーアプリケーション
クライアントアプリケーション
109 ネットワーク
103 クライアントシステム
104
クライアントアプリケーションバージョン管理装置
105 ディスク装置
クライアント
アプリケーション
アプリケーション
定義ファイル
106 クライアントシステム
107
クライアントアプリケーションバージョン管理装置
108 ディスク装置
クライアント
アプリケーション
アプリケーション
定義ファイル

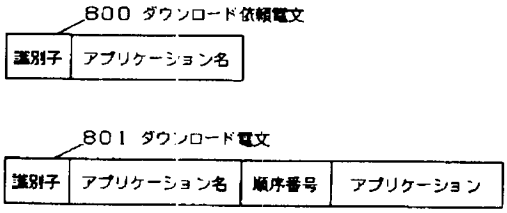
【図6】


開 始
6・1
バージョン検査依頼電文受信
6・2
バージョン比較部へファイル名と作成日時を渡す
6・3
ディスク装置に保持しているクライアントアプリケーションから作成日時取得
6・4
作成日時は一致しているか
YES
NO
結果フラグにバージョン一致を設定
結果フラグにバージョン不一致を設定
6・5
6・6
検査依頼応答電文送信
終 了

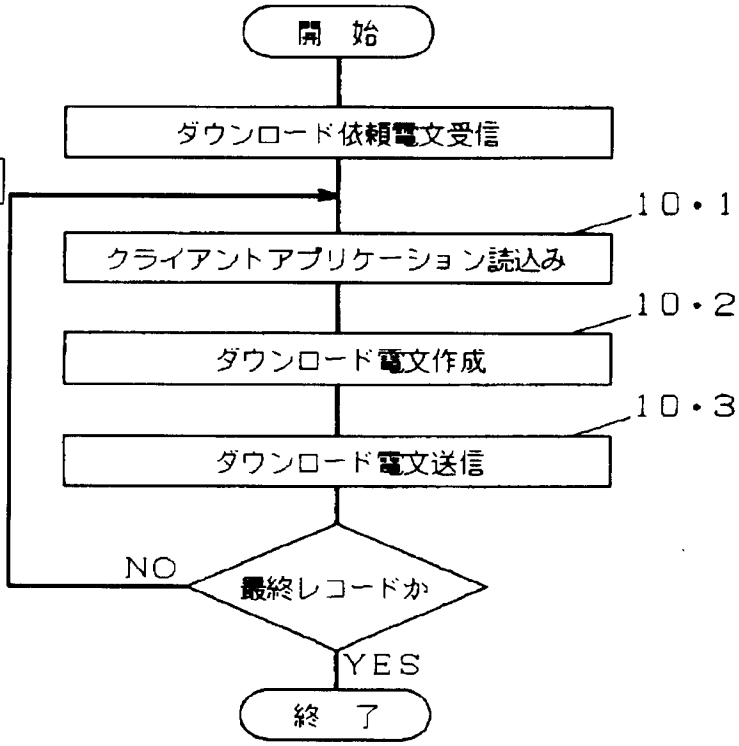
【図7】


71 サーバーアプリケーションバージョン管理装置
201
バージョン比較部
700
ダウンロード部
202
電文処理部
203
送受信部
200 ディスク装置
クライアントアプリケーション
205
電文処理部
207
送受信部
208
ファイル名格納領域
206
ファイル管理部
701
アップデート部
204 ディスク装置
クライアントアプリケーション
アプリケーション定義ファイル
72 クライアントアプリケーションバージョン管理装置

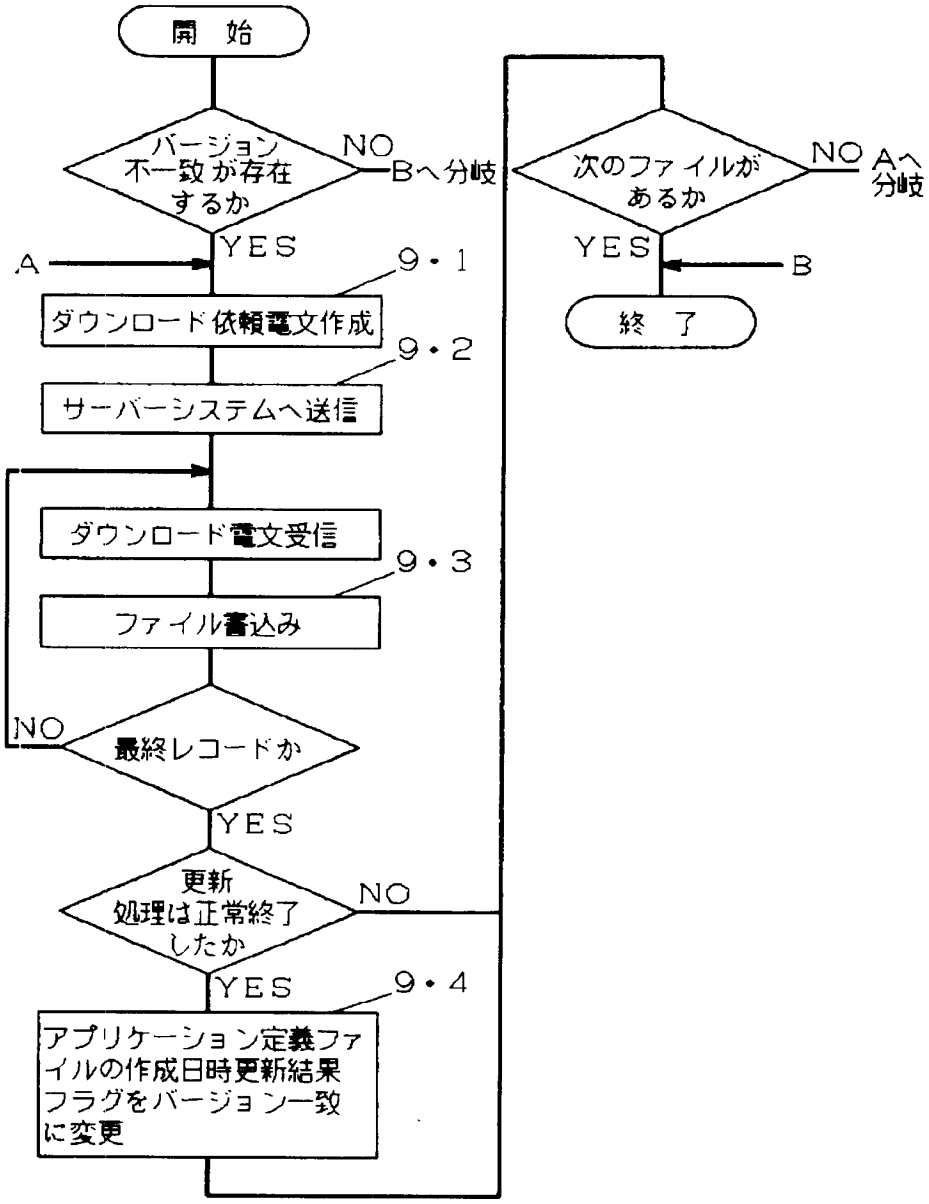
【図8】


800 ダウンロード依頼電文
識別子
アプリケーション名
801 ダウンロード電文
識別子
アプリケーション名
順序番号
アプリケーション

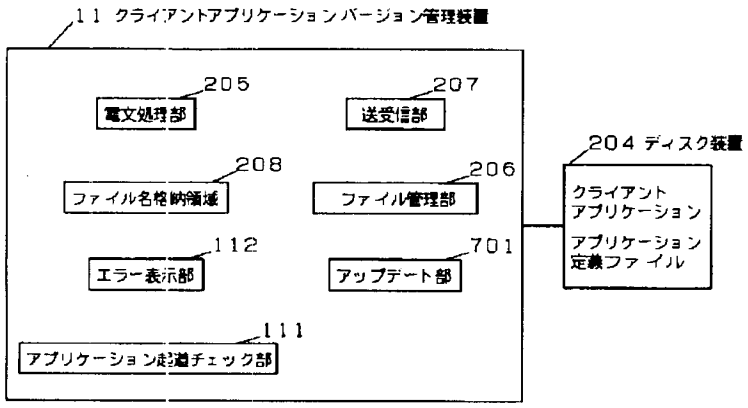
【図10】


開 始
ダウンロード依頼電文受信
10・1
クライアントアプリケーション読込み
10・2
ダウンロード電文作成
10・3
ダウンロード電文送信
NO
最終レコードか
YES
終 了

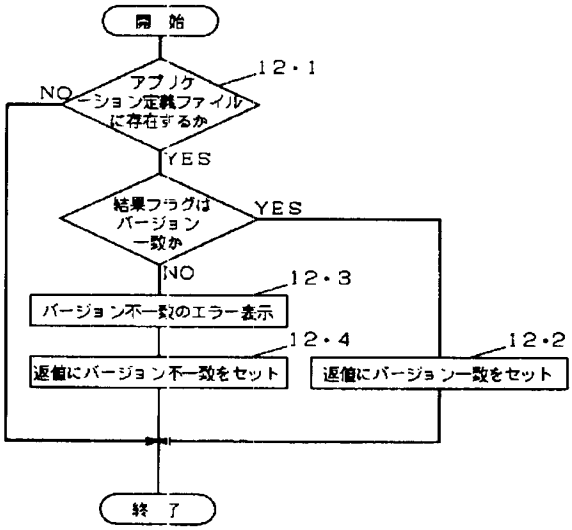
【図9】


開　始
バージョン不一致が存在するか
NO
Bへ分岐
YES
A
9・1
ダウンロード依頼電文作成
9・2
サーバーシステムへ送信
ダウンロード電文受信
9・3
ファイル書込み
NO
最終レコードか
YES
更新処理は正常終了したか
NO
YES
9・4
アプリケーション定義ファイルの作成日時更新結果フラグをバージョン一致に変更
次のファイルがあるか
NO
Aへ分岐
YES
B
終　了

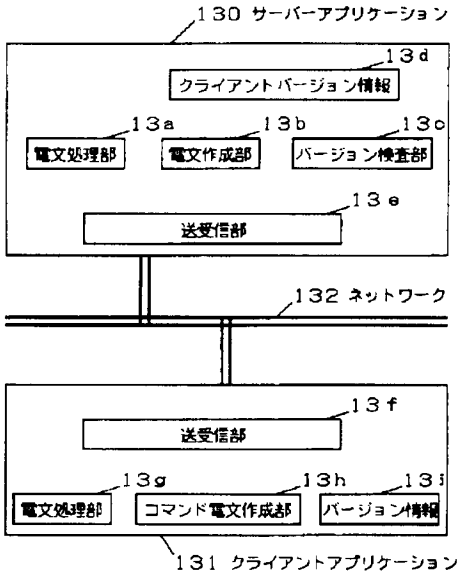
【図１１】


11 クライアントアプリケーションバージョン管理装置
205
電文処理部
207
送受信部
208
ファイル名格納領域
206
ファイル管理部
112
エラー表示部
701
アップデート部
111
アプリケーション起動チェック部
204 ディスク装置
クライアント
アプリケーション
アプリケーション
定義ファイル

【図１２】


開 始
アプリケーション定義ファイルに存在するか
12・1
NO
YES
結果フラグはバージョン一致か
YES
NO
12・3
バージョン不一致のエラー表示
12・4
返値にバージョン不一致をセット
12・2
返値にバージョン一致をセット
終 了

【図１３】


130 サーバーアプリケーション
13d
クライアントバージョン情報
13a
電文処理部
13b
電文作成部
13c
バージョン検査部
13e
送受信部
132 ネットワーク
13f
送受信部
13g
電文処理部
13h
コマンド電文作成部
13i
バージョン情報
131 クライアントアプリケーション